Hypercubic Combinatorics: Hamiltonian Decomposition and Permutation Routing*

William Duckworth
Mathematical Sciences Institute,
Australian National University,
Canberra, ACT 0200, Australia.

Alan Gibbons
Department of Computer Science, King's College,
London, WC2R 2LS, UK.

Abstract

In this paper we first present new proofs, much shorter and much simpler than can be found elsewhere, of two facts about Hypercubes: that for the d-dimensional Hypercube, there exists sets of paths by which any *permutation routing* task may be accomplished in at most $2d - 1$ steps without queueing and, when d is even, there exists an edge decomposition of the Hypercube into precisely $d/2$ edge-disjoint Hamiltonian cycles. The permutation routing paths are computed *off-line.* Whether or not these paths may be computed by an *on-line* parallel algorithm in $O(d)$-time has long been an open question. We conclude by speculating on whether the use of a Hamiltonian decomposition of the Hypercube might lead to such an algorithm.

1 Introduction

The great importance of the Hypercube as an *interconnection network* for parallel computers has stimulated a great deal of research into its structural properties [7] and its ability to support efficient parallel routing algorithms. Routing issues are reviewed in [6, 9] and we outline these in Section 4.

*An abridged version of part of this work appeared in the Proceedings of the 9^{th} Australasian Workshop on Combinatorial Algorithms (AWOCA'98)

Some current proofs of hypercubic properties and related algorithmics are, however, in a form that begs refinement and simplification. For example, the fact that the d-dimensional Hypercube has sets of paths by which any *permutation routing* task may be accomplished in at most $2d - 1$ steps without queueing is normally established by a lengthy proof involving the Beneš network (e.g. in [9]). In Section 2 we show that a much simpler and direct proof of this fact is possible.

The problem of permutation routing on an interconnection network is the problem of sending one (constant length) message from each processor (Hypercube node) in such a way that each processor receives one message. In any one instance, the initial address and final address of the messages define a *permutation* of the node labels and an algorithm should work for any permutation. The algorithm we describe in Section 2 is determines, *off-line*, which set of paths may be used for any prescribed permutation. Vöcking [13] recently introduced faster routing algorithms, however, these are randomised algorithms whereas our algorithm is deterministic.

Similarly, it seems to be well-known fact that every Hypercube with even degree has an exact decomposition into edge-disjoint Hamiltonian cycles. There does not appear, however, to be an *explicit* proof of this fact in the literature. The fact does follow from the consideration of the Hamiltonian decomposition of general graphs [1]. These considerations are far more complicated that they need to be for the special case of Hypercubes.

In Section 3 we describe a proof, specifically tailored to Hypercubes of even degree, that an edge-disjoint Hamiltonian decomposition is possible. In Section 4, we speculate on how such a decomposition of the Hypercube might be employed to provide an improved on-line deterministic algorithm for the permutation routing problem.

Recall that a *Hypercube* is a graph with $n = 2^d$ nodes, where d is a non-negative integer called the *dimension* of the Hypercube. If the nodes are labeled in binary from 0 to $n - 1$, then there is an edge between two nodes if and only if their labels differ in precisely one binary bit. If an edge joins two nodes whose addresses differ in their i^{th} bit, then the edge is said to belong to the i^{th} *dimension*. We consider the edges of the Hypercube to be *bi-directional*, such that two messages may be swapped between adjacent nodes in one parallel step.

2 Permutation Routing on the Hypercube

In this section we provide a simple proof of the following theorem.

Theorem 1 *For any one-to-one mapping of $n=2^d$ (constant length) inputs to n outputs on a d-dimensional Hypercube, all of the n inputs can be routed in parallel to their destinations in at most $2d - 1$ steps without queueing.*

Proof The proof is by induction on the dimension d of the Hypercube. When $d = 1$ the Hypercube consists of a single edge which connects two processor sites. In this case at most one routing step is required to deliver the messages and so we have a basis for our induction.

For the induction Hypothesis, assume that the theorem is true for all Hypercubes with dimension less than $d > 1$. We now prove the inductive step that the theorem must also be true for the Hypercube H_d of dimension d. We consider H_d as two copies of H_{d-1} connected by edges of the d^{th} dimension. Figure 1 illustrates this. Each node is also associated with a package to be routed (in general) to another node. If we suppress the d^{th} bit of both the node and destination addresses of the packages, we see that for each binary address from 0 to $(n/2) - 1$, H_d has exactly two nodes with identical addresses (one at either end of an edge of dimension d) and two packages with identical destination addresses. We now show, by the following off-line computation, that in one parallel step and using only edges of the d^{th} dimension, we can route all packages so that each copy of H_{d-1} contains exactly one package with any one destination address.

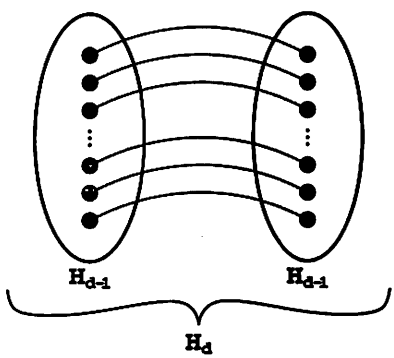

Figure 1: H_d represented as two copies of H_{d-1}

Let (m_i, m_j) represent an edge of the d^{th} dimension where the left node of this edge contains a package with destination address m_i and the right node contains a package addressed to m_j, where i and j represent addresses. Notice that i and j are not necessarily distinct. Consider every edge of the d^{th} dimension, for which i and j are distinct, in the following sequence. Let the first edge be (m_i, m_j). Now choose the next edge to be the other edge containing m_j, within our notation this will be either (m_j, m_k) or (m_k, m_j). In the former case leave packages where they are and in the latter case swap the messages.

The subsequence of edges chosen is now $((m_i, m_j), (m_j, m_k))$. If $i = k$, we choose a new edge arbitrarily and start a new subsequence otherwise we continue the current subsequence by considering the other edge containing m_k in the same way that the edge (m_i, m_j) was considered. In this way, after all edges have been considered, we will have ensured that each copy of H_{d-1} contains exactly one package with any one destination address and we will have achieved our objective.

In one parallel routing step we have now reduced our initial problem to two smaller instances of the permutation routing problem on Hypercubes of dimension $d-1$. By the inductive hypothesis, these problems can be solved in $2(d-1)-1$ parallel steps. After their solution, we now simply restore the d^{th} bit of the node and destination addresses. Now any package will be either at its destination or at the node adjacent to its destination along the edge of dimension d. Thus after one more parallel routing step every message will be at its destination.

The total number of parallel routing steps for any permutation routing problem on H_d is therefore $1 + (2(d-1)-1) + 1$ which is $2d-1$. This completes the proof of the number of steps claimed by the algorithm. Since each routing step consists of a number of (parallel) package swaps on edges forming a subset of a perfect matching of H_d, it follows that no queueing will occur when routing packages in parallel to the destinations. □

Baumslag and Annexstein [3], also give a similar proof that, for all Cartesian product networks, any permutation routing task can be achieved in at most $2d$-1 steps without queueing. Since the Hypercube is also a Cartesian product network, their proof also applies to Hypercubes. We still believe, however, that for the specific case of Hypercubes, our proof is simpler and may be more suitable in pedagogic settings.

3 Hamiltonian Decomposition of Hypercubes

In this section we prove that, for even d, there exists a decomposition of H_d into $d/2$ edge-disjoint Hamiltonian cycles. That this is true follows from general results on Hamiltonian decomposition of graphs and involves extended and unnecessarily tedious proofs (see [1, 2, 5]) for the Hypercube. A direct short proof for Hypercubes has not previously appeared in the literature. Theorem 2 contains our proof which proceeds by induction on the dimension of the Hypercube. Before presenting that theorem, we need to establish Lemmas 1 and 2.

The Cartesian product $G \times G'$ of two graphs $G = (V, E)$ and $G' = (V', E')$ is defined to be the graph with vertex set $V(G) \times V(G')$ and with an edge between (x_i, y_i) and (x_j, y_j) whenever ($x_i = x_j$ and $(y_i, y_j) \in E'$) or ($y_i = y_j$ and $(x_i, x_j) \in E$).

Lemma 1 and Lemma 2 are simplifications of two results of Kotzig [8] and in what follows C_n denotes the cycle of length n.

Lemma 1 *For even m and n, the edge set of the Cartesian product of C_m and C_n can be partitioned into two edge disjoint Hamiltonian cycles.*

Proof The proof is by induction on the cycle lengths. We take the basis for the induction to be $C_4 \times C_4$. This decomposition is shown in Figure 2.

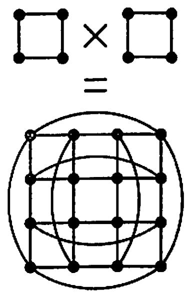

Figure 2: Decomposition of $C_4 \times C_4$ into two Hamiltonian cycles

Inductively assume that $G = C_{m-2} \times C_n$ can be decomposed into two Hamiltonian cycles, C^1 and C^2. We now show that $G' = C_m \times C_n$ can be decomposed into two Hamiltonian cycles, $C^{1'}$ and $C^{2'}$. Let $V(G) = \{v_{i,j} | 1 \le i \le n, 1 \le j \le m-2\}$ and $V(G') = \{v'_{i,j} | 1 \le i \le n, 1 \le j \le m\}$. G consists of an $n \times (m-2)$ grid of vertices. Each row of $m-2$ vertices is exactly one copy of C_{m-2} and each column of n vertices is exactly one copy of C_n (see Figure 3).

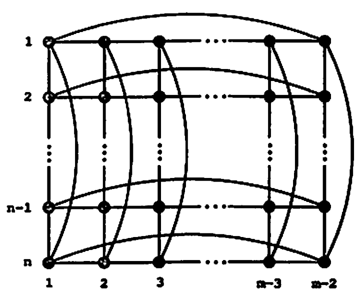

Figure 3: $C_{m-2} \times C_n$

G' can be generated by removing the edges $(v_{i,m-3}\ ,\ v_{i,m-2})$ in each of the cycles of C_{m-2} in G. Insert two new copies of C_n into this space and label the columns j' and j''. Now replace the missing edges to complete the cycles.

For each edge that was removed from each copy of C_{m-2}, add the edges $(v_{i,m-3}\ ,\ v_{i,j'})$ and $(v_{i,j''}\ ,\ v_{i,m-2})$. Now connect each pair of these added edges in the following way. If there has been an edge removed on row i and an edge removed on row $i+1$, then connect the two new edges on row i by adding an edge from $(v_{i,j'}\ ,\ v_{i,j''})$. Otherwise continue to add edges of the form $(v_{i,j'}\ ,\ v_{i+1,j'})$ and $(v_{i,j''}\ ,\ v_{i+1,j''})$ (read all values of i modulo n) until the row $i+x$ has a removed edge where an edge $(v_{x-1,j'}\ ,\ v_{x-1,j''})$ is added to complete the cycle. This can be illustrated as follows.

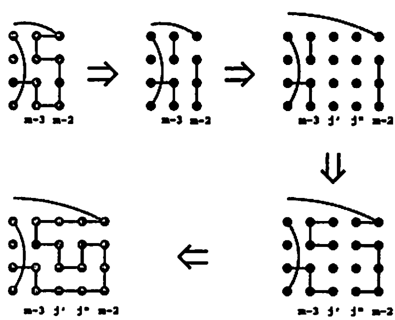

Figure 4: Extending a cycle of $C_{m-2} \times C_n$ to give a cycle of $C_m \times C_n$

The result is the two Hamiltonian cycles of $C_m \times C_n$ as required. □

In fact the Cartesian product of C_m and C_n is decomposable into two Hamiltonian cycles whatever the positive integer values of m and n [2, 5]. For our purposes, we need only establish the Lemma for m and n even.

Lemma 2 *The Cartesian product of a graph G (which is decomposable into 2 Hamiltonian cycles) and a single cycle is decomposable into 3 Hamiltonian cycles.*

Proof This proof is a simplification of that found in [2] and proceeds by induction on the number of edges in the cycles. As a basis for our induction we use the Cartesian product of H_4 (decomposed into two cycles) and the single cycle C_4. Figure 5 shows how H_4 can be decomposed into two edge disjoint Hamiltonian cycles [5].

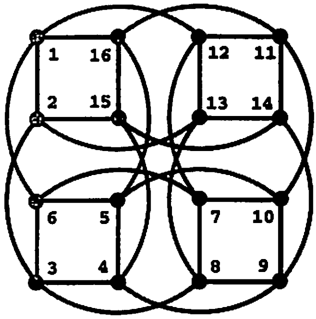

Figure 5: Edge decomposition of H_4 into two Hamiltonian cycles

Using this decomposition of H_4, the Cartesian product of $H_4 \times C_4$ is illustrated in Figure 6.

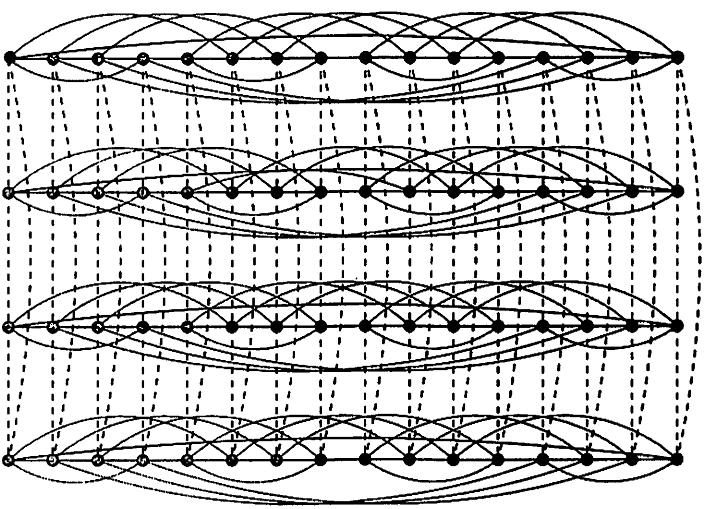

Figure 6: The Cartesian product of $H_4 \times C_4$

The blue edges represent copies of one of the cycles in H_4, and the red edges represent the other cycle. The dashed edges are copies of C_4.

The first step is to number the vertices of H_4 around one of the cycles from 1 to 16 starting at some arbitrary point (see Figure 5). The cycle that remains has two vertices connected to the node labeled '1'. Name these vertices a and b, where $a < b$. In the example, $a = 6$ and $b = 12$.

Construct a Hamiltonian cycle of H_6 in the following way.

- Make 4 copies of the unnumbered cycle H_4 (call them A, B, C, D)
- Remove the edges (1, a) from each copy A, B, C, D
- Remove the edges (1, b) from each copy except the first and the last
- Add the edges (A1, B1) , (B1, C1) , (C1, D1)
- Add the edges (Aa, Ba) , (Bb, Cb) , (Ca, Da)

The notation (A1,B1) represents an edge between vertex 1 in graph A and vertex 1 in graph B. The resulting graph is a cycle and looks as follows:

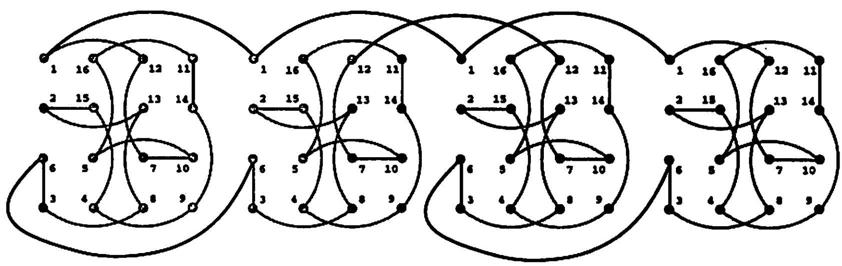

Figure 7: A complete cycle

Then remove this complete cycle from the graph product. The rest of the graph product consists of a grid of sixteen columns of four vertices and is shown in Figure 8.

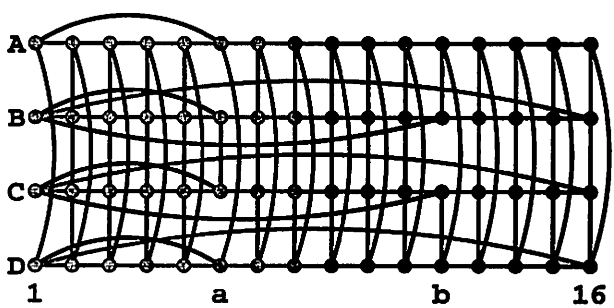

Figure 8: Remainder of graph product with one cycle removed

The next step is to remove columns of four vertices from the remainder of the graph product (two columns at time) in such a way to ensure: (*i*) the value of a is reduced to 4; (*ii*) the value of b is reduced to 6; and (*iii*) the length of the product graph is reduced to total of 8 vertices.

The node that is labeled '1' was chosen arbitrarily. Consider the binary representation of the nodes of the Hypercube. The chosen node can, without loss of generality, represent node 0000. This node in H_4 is connected to four other nodes, namely 0001, 0010, 0100 and 1000. Choosing any pair of these nodes to be connected to node 0000 to form a cycle, 0001 and 1000 say, leaves the other pair to form the other cycle. The minimum distance between any 2 of these four nodes is two. The numbering of one of these cycles from 1, 2, ..., 16, gives minimum and maximum values for the nodes a and b.

One cycle is of the form 0000, 0001, ..., 1000. This is the cycle that is numbered 1, 2, ..., 16. In order to connect 0001 to 0100 or 0010 the minimum distance is two, so the minimum value of a is 4. Node b is represented by 0100 or 0010 also, which must be of minimum distance 2 from 1000 which gives b a maximum value of 14. The convention of naming a and b such that $a < b$ means that if the maximum value of b is 14, then the maximum value of a is 12 and if the minimum value of a is 4 then the minimum value of b is 6.

This also shows that the values of a and b must always be even. As the nodes labeled 2 and 16 are represented by even numbers and to connect either of these two nodes to either of the two other possible a and b nodes, must involve a distance which is even, means that the values of a and b must always be even.

By removing numbers of copies of C_4 in pairs from between any number of the following locations: (*i*) 1 and a; (*ii*) a and b; and (*iii*) b and 16, it is always possible to reach the base case where $a = 4$, $b = 6$ and the length of the graph product is 8.

In the example, the values of a and b are 6 and 12 respectively, therefore removing the dashed columns in Figure 9 will produce the desired result.

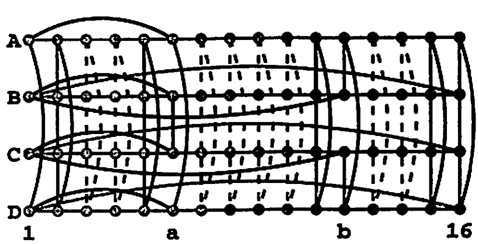

Figure 9: Remainder of graph product with one cycle removed

Observe how the removal of these edges does not affect the cycle already constructed and also the connecting edges to vertices a and b. Shrink the remaining edges to generate the graph shown in Figure 10. This graph can be decomposed into the 2 cycles as shown.

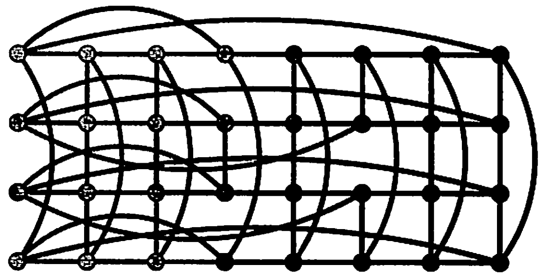

Figure 10: The base case

These two cycles can be extended to the original size of the product of $H_4 \times C_4$ by replacing the same number of columns in the graph that were removed and returning them to the places that they were taken from. The method of extending the cycles is the same as that shown for Lemma 1. The resulting 2 cycles and the cycle constructed by the method above are the three Hamiltonian cycles of $H_4 \times C_4$.

For any graph G which is decomposable into two cycles, the method of construction of the first cycle in the product $G \times C_4$ remains the same. When considering the specific case that G is the edge disjoint union of two Hamiltonian cycles from a Hypercube, the base case that is used for the construction of the first cycle in the product graph is always attainable due to the connectivity of the Hypercube described earlier.

Assume that $G \times C_{n-2}$ can be decomposed into three Hamiltonian cycles (where G is decomposable into two Hamiltonian cycles). $G \times C_n$ can be decomposed into three Hamiltonian cycles. The principle is the same as that for the base case, except that now, when the first cycle has been constructed and removed, and the remainder of the graph product reduced, the resulting graph is shown by Figure 11 and can be decomposed into the cycles as shown.

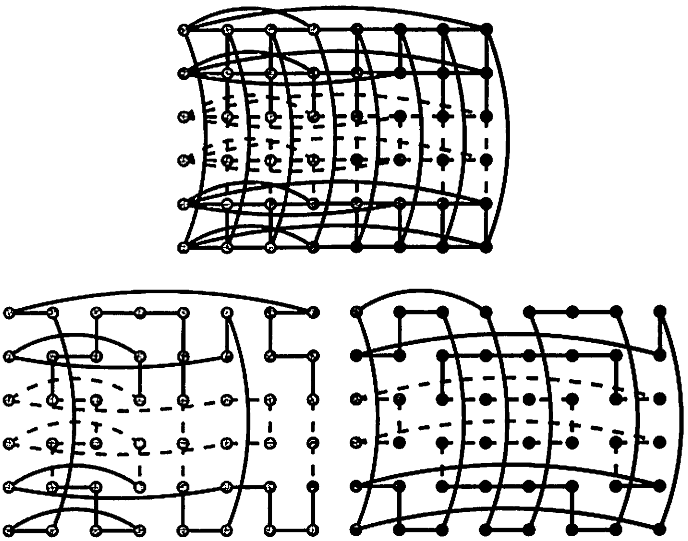

Figure 11: Remove dashed edges to reduce graph product

The vertices between 2 and $n-1$ can be constructed from multiple copies of the rows x and y and their incident edges giving products of cycles of any even length $\geq$ 4. □

Theorem 2 *The 2d-dimensional Hypercube is decomposable into d edge-disjoint Hamiltonian cycles.*

Proof The proof is by induction on d. As the basis for our inductive proof we choose $d = 2$ or $d = 3$, then H_4 and H_6 are decomposable into two and three edge disjoint Hamiltonian cycles respectively as we have already shown. We assume that the theorem holds for all dimensions less than $2d > 6$, and prove the theorem holds for H_{2d}. We represent the Hypercube H_{2d} in the following way:

$$H_d \times H'_d \text{ where } d \text{ is even or } H_{d+1} \times H_{d-1} \text{ where } d \text{ is odd.}$$

In both cases H_{2d} is the Cartesian product of two smaller Hypercubes of even dimension. By the induction hypothesis, this satisfies the theorem and these Hypercubes have Hamiltonian decompositions. In the first case, where d is even, H_d can be represented as $C^1 \bigcup_e C^2 \bigcup_e \dots \bigcup_e C^{\frac{d}{2}}$ and H'_d can be represented as $C'^1 \bigcup_e C'^2 \bigcup_e \dots \bigcup_e C'^{\frac{d}{2}}$ where C^i or C'^i represents a Hamiltonian cycle of H_d or H'_d respectively, and $\bigcup_e$ represents the edge disjoint union of cycles.

H_{2d} can therefore be expressed in the following way

$$(C^1 \times C'^1) \bigcup_e (C^2 \times C'^2) \bigcup_e \ldots \bigcup_e (C^{\frac{d}{2}} \times C'^{\frac{d}{2}}).$$

Each term is the Cartesian product of two cycles and from Lemma 1, each of these gives two Hamiltonian cycles of H_{2d} giving a total of $\frac{d}{2} \times 2 = d$ cycles.

In the second case, where d is odd, H_{d-1} can be represented as

$$C^1 \bigcup_e C^2 \bigcup_e \ldots \bigcup_e C^{\frac{d-1}{2}}$$

and H_{d+1} can be represented as

$$C'^1 \bigcup_e C'^2 \bigcup_e \ldots \bigcup_e C'^{\frac{d-1}{2}} \bigcup_e C'^{\frac{d+1}{2}}$$

where C^i represents a Hamiltonian cycle of H_{d-1} and C'^i represents a Hamiltonian cycle of H_{d+1}. H_{2d} can therefore be expressed as follows.

$$(C^1 \times C'^1) \bigcup_e (C^2 \times C'^2) \bigcup_e \ldots \bigcup_e (C^{\frac{d-3}{2}} \times C'^{\frac{d-3}{2}}) \bigcup_e (C^{\frac{d-1}{2}} \times (C'^{\frac{d-1}{2}} \bigcup_e C'^{\frac{d+1}{2}})).$$

The first $\frac{d-3}{2}$ terms are the Cartesian product of two cycles, each giving two Hamiltonian cycles of H_{2d}, and the final term is the Cartesian product of a single cycle and the edge disjoint union of two Hamiltonian cycles which, by Lemma 2, gives us the remaining three Hamiltonian cycles giving a total of $(\frac{d-3}{2} \times 2) + 3 = d$ cycles and so theorem follows. □

4 Open Problems

A lower bound for the running time of permutation routing algorithms on the Hypercube is $O(\log n)$, provided by the diameter of the network. It is not known at present whether there exists a distributed $O(\log n)$-time *deterministic* algorithm for permutation routing on the Hypercube. That $O(\log n)$-time permutation routing is at all possible was discovered around 1980. The celebrated two-phase *randomised* routing approach was first conceived by Valiant and developed by Valiant and Brebner [10, 11, 12].

As far as deterministic permutation routing is concerned, at least for all practical purposes, the best known algorithms run in $O(\log^2 n)$ time. Several such algorithms are based upon simulations of the classical sorters of Batcher including, for example, the bitonic sorter.

At this time, the fastest known deterministic algorithm for permutation routing on the Hypercube is due to Cypher and Plaxton [4] which runs in $O(\log n(\log\log n)^2)$ time or $O(\log n(\log\log n))$ time with a substantial amount of off-line computation. Asymptotically, this is an improvement over the the $O(\log^2 n)$ algorithms already mentioned, however, because of the large constants hidden by the notation, Cypher and Plaxton's algorithm only becomes competitive for Hypercubes of dimension greater than 20. The algorithm is extremely intricate but is well described in [9].

There is a continuing strong interest to answer the difficult question whether an $O(\log n)$ time deterministic algorithm exists for permutation routing on the Hypercube. It is interesting to observe that no impediment arises from the need to have $O(\log n)$ length paths for any permutation such that, in any given parallel routing step, no two messages occupy the same edge. Such path sets are established by the *off-line* algorithm described in Section 2. It is interesting to ask, as seems likely, whether such sets of paths might also be established through defining (in each instance) a sequence of $O(d)$ (directed) Hamiltonian circuits by which, in any one time step, all messages are sent along one out-edge defined by the circuit. This is dual to the notion of using edges of *matchings* which is what the sets of paths defined by the algorithm of Section 2 employs.

If such sets of paths can be defined in this way, it is then an intriguing question as to whether the $d/2$ edge-disjoint Hamiltonian circuits, given by a decomposition such as that described in Section 3, is a rich enough set to provide this. There are, of course, an exponentially large number of different Hamiltonian circuits in any Hypercube, but such a set is likely to provide too rich a selection for our problem. Notice that the decomposition into Hamiltonian circuits of Section 3 provides an element of choice in what Hamiltonian Circuits appear in the decomposition and this selection might usefully be driven by the particular (permutation) routing problem in hand.

It is likely that the open problems just described will have not too difficult positive resolutions for permutation routing. If this is so, we will then be left to resolve the perhaps more difficult questions as to whether they have *on-line* $O(d)$ distributed implementations.

We believe that this paper, apart from presenting new short proofs of hypercubic properties (which will no doubt be of especial benefit in pedagogic situations), poses some interesting lines of research in terms of the questions just posed. Such questions can, of course, be easily extended to more general routing problems (i.e. other than permutation problems) on the Hypercube.

References

[1] B. Alspach, J.C. Bermond and D. Sotteau. Decomposition Into Cycles I: Hamilton Decompositions. Technical Report TR 8712, Department of Mathematics and Statistics, Simon Fraser University, Canada, 1987.

[2] J. Aubert and B. Schneider. Decomposition de la Somme Cartisienne d'un Cycle et de l'union de Deux Cycles Hamiltoniens en Cycles Hamiltoniens. Discrete Mathematics, 38:7–16, 1982.

[3] M. Baumslag and F. Annexstein. A Unified Framework for Off-Line Permutation Routing in Parallel Networks. Mathematical Systems Theory, 24:233–251, 1991.

[4] R. Cypher and C.G. Plaxton. Deterministic Sorting in Nearly Logarithmic Time on the Hypercube and Related Computers. Proceedings of the 22^{nd} ACM STOC, 193–203, 1990.

[5] M.F. Foregger. Hamiltonian Decompositions of Products of Cycles. Discrete Mathematics, 24:251–260, 1978.

[6] A. Gibbons and P. Spirakis (editors). Lectures on Parallel Computation. Cambridge International Series on Parallel Computation, Volume 4, Cambridge University Press, 2005.

[7] F. Harary, J.P. Hayes and H.J. Wu. A Survey of the Theory of Hypercube Graphs. Computers and Mathematics with Applications, 15(4):277–289, 1988.

[8] A. Kotzig. Every Cartesian Product of two Circuits is Decomposable into two Hamiltonian Circuits. Technical report 233, Centre de Recherches Mathematique, Montreal, 1973.

[9] F.T. Leighton. Introduction to Parallel Algorithms and Architectures: Arrays, Trees and Hypercubes. Morgan Kaufmann, 1992.

[10] L. G. Valiant and G. J. Brebner. Universal Schemes for Parallel Communication. Proceedings 13^{th} ACM STOC, 263–277, 1981.

[11] L. G. Valiant. Experiments with a Parallel Communication Scheme. Proceedings of the 18^{th} Allerton Conference on Communication, Control and Computing, 861–811, 1980.

[12] L. G. Valiant. A Scheme for Fast Parallel Communication. SIAM Journal of computing, 11:350–361, 1982.

[13] B. Vöcking. Almost Optimal Permutation Routing on Hypercubes. Proceedings of the 33^{rd} ACM STOC, 530–539, 2001.